Example of Enhanced Prompt for Generating Python Code

Enhanced Prompt for Generating Python Code

Objective: Develop a Python script that reads data from an Excel file, performs regression analysis, and calculates the number of futures contracts needed to hedge a portfolio.

Requirements:

Excel File Details:

Filename: Example_PortfolioManager.xlsx

Sheets:

-Sheet 1: S&P500 futures

      -Column B (Second Column): Futures Prices

      -Column C (Third Column): Futures Returns

-Sheet 2: Portfolio

      -Column B (Second Column): Portfolio Values

      -Column C (Third Column): Portfolio Returns

Data Processing Steps:

Data Importation:

-Read the specified Excel file.

-Extract futures prices and returns from the S&P500 futures sheet.

-Extract portfolio values and returns from the Portfolio sheet.

Data Alignment:

-Ensure that the returns and prices are properly aligned by date or corresponding periods.

-Handle any missing or inconsistent data appropriately (e.g., by dropping incomplete rows).

Regression Analysis:

-Perform an Ordinary Least Squares (OLS) regression of the portfolio's returns on the S&P500 futures returns.

-Include a constant term in the regression model.

-Extract the regression coefficient (slope) of the S&P500 futures returns and assign it to a variable named beta.

-Extract the R-squared value of the regression and assign it to a variable named Effectiveness.

Hedging Calculations:

-Retrieve the last available futures price from the S&P500 futures sheet and assign it to a variable named FutPrice.

-Retrieve the last available portfolio value from the Portfolio sheet and assign it to a variable named PortPrice.

-Calculate the number of contracts needed to hedge the portfolio using the formula:

Number of Contracts=PortPrice(250*$FutPrice)$*1,000,000

Assign the calculated number of contracts to a variable named number_of_contracts.

Output:

Display or print the following variables with appropriate formatting:
-beta
-Effectiveness
-FutPrice
-PortPrice
-number_of_contracts

Technical Specifications:

-Programming Language: Python
-Libraries to Use:
    -pandas for data manipulation and Excel file reading.
    -statsmodels for performing regression analysis.

-numpy for numerical operations (if needed).

-Error Handling:

-Include error handling for scenarios such as missing files, incorrect sheet names, or missing columns.

-Code Documentation:

-Add comments explaining each major step for clarity and maintainability.

-Assumptions:

-The Excel file is well-structured with no merged cells or hidden rows/columns in the relevant sheets.

-The date alignment between the S&P500 futures and Portfolio sheets is consistent and does not require additional synchronization.